

QL TECHNICAL REVIEW

ISSUE 5

PRICE £1.50

CONTENTS

EDITORIAL	2
NEWSETTES	3
TURBOCHARGE - REVIEW	4 - 12
MORE BITS AND PIECES	12
MACHINE CODE UNDER QDOS	13 - 16
DILWYN JONES COMPUTING PROGS - REVIEW	17 - 19
SUPER TOOLKIT II - PART 2	20 - 25
MODE CHANGE UTILITY	25
LOOSE ITEMS	26, 34
ST - QL UPDATE	26
FLASHBACK - REVIEW	27 - 29
QMENUS - NOTES	30
THE PAINTER - REVIEW	31 - 32
CLIP ART - REVIEW	33 - 34
COCK-UP CORNER	34
BROTHER HR-5 UPDATE	35
MANDELBROT UPDATE	36
P.D. NEWS	36

PUBLISHED BY C.G.H. SERVICES,
CWM GWEN HALL, PENCADER, DYFED,
CYMRU, SA39 9HA

EDITORIAL

Firstly, thanks to everyone who has been sending in material for us to print. We actually have enough to fill both this issue and number 6 already in hand, and there's more on the way! Keep things coming in. (Don't forget to put your name at the end of any articles you send in!)

The way QLTR is shaping it looks as though we shall be seeing more items on machine code, combining QL's and printers, using the various Toolkits and Compilers, reviews of business programs and databases to name but a few.

I'd like to see more by way of beginners' hints on SuperBasic as this language is one that all QL users potentially have in common, and with the excellence of the two major compilers there's no reason not to use SuperBasic for items that do not depend on very fast code.

We have a few special items in this and next issue, in particular Rich Mellor has kindly translated a few items from QUASAR, the excellent magazine from the Sinclair QL User Club of Germany. We have two more issues of this magazine awaiting translation of interesting articles. Let us know if you fancy having go at them.

Our first attempt at dealing with disks and disk drives was not an unqualified success, see cock-up corner for some details, so next issue will see a second go at it. We'll be incorporating some of the information that Dennis Briggs and Tony Firshman have sent in. More welcome, of course!

There has been some criticism of our layout. Please keep in mind that we do not have unlimited funds for the purchase of software and hardware. At present we use Quill for the processing of all document files and we then print them out in our set lengths and widths. As the QL is not the best machine I have I do not want to invest in more word processors for it. As for the headings they are produced on a second-hand ST (1 meg) using a second-hand Publishing Partner. As yet there is no likelihood that I will be able to combine all this on one machine and in one program. Producing this magazine is quite time-consuming. To add to it by having to learn new ways of doing things will not help. Our aim is to be readable and interesting. Selling only 300 copies is not a base from which to embark on massive investment in Desk-Top Publishing!

So, even though you may have a brilliant word processor on your machine, I can do everything I need with Quill (plus TurboQuill+, Spellbound etc). Articles are more likely to get printed if I do not have to import them into Quill line by line and reformat each line of text.

There has also been some criticism that we do not necessarily give publishers the opportunity to reply to reviews prior to publication. If we get time, we do send reviews to publishers, if we feel that the review will prove contentious or makes criticisms that are possibly incorrect. Correction of errors should accompany any review. However more subjective comment, by its nature, cannot be corrected, only differing views expressed. Also we have limited space in the mag and limited time to produce each issue.

In an attempt to keep QLTR current we are instituting an "Error Report" which will be supplied free to subscribers. The purpose of this is to inform people of substantive changes from information published in earlier issues of QLTR. These may range from version updates, changes of address, "pointers" to corrections published in later issues etc. I suspect that if we keep going long enough the "Error Report" will get bigger than the magazine itself!

Finally a special thank you to everyone who has helped us with our sales stalls at various computer shows and workshops recently, also to everyone who has spoken to us at these shows. My apologies if anyone has not always received the most intelligent of responses from myself at these events, the pressure of the moment, not to mention the lack of refreshment facilities and the distraction from other "customers" may well have led to a substantial response degradation. (What?)

Right, it only remains to say that the deadline for #7 is Feb 1st, 1991. We look forward to hearing from you! (The shorter the letter, the quicker the reply!)

Do let us know if you like the idea of an optional "cover-disk" for QLTR containing programs, screen dumps and recommended PD progs etc. I think there's enough material for one to cover this and next issue of QLTR. Cost - about £1.50 all incl.

Richard Alexander

NEWSETTES

QUANTA have reported that PDQL are still loading. Reports of problems with them to the Consumer Protection Office, 155 - 157 Corporation Street, Birmingham, B4 6TH. Both David Walker and Joe Haftke are removing their programs from PDQL. QL World are not taking any more adverts.

MINERVA ROMS

Simon Goodwin has had a look at problems with Minerva in some of its manifestations in the December QL World. At the recent All Formats Show in London I had a discussion about Minerva with Dilwyn Jones and Freddy Vachha, and have also chatted about it with Robin Barker of Di-Ren. The consensus we seem to be approaching is that we'd rather QView stopped releasing new versions of Minerva for the time being and settled on a "bug-free" version, continuing development but not releasing them. This would mean that we no longer have a "moving target" to aim at and could concentrate on making our products compatible (where possible) with just one version of Minerva. At the moment there are so many versions around none of us know which we should be trying to be compatible with. The latest version available is 1.84.

Digital Precision have advertised their new word processor in the December 1990 QL World. As this is by Steve Sutton we can be almost certain that it'll be an excellent product. If anyone buys a copy we'd love to have a review.

○ have released Polytext, written by Nick Ward. The exact capabilities will depend on your printer, best to write in for details. The basic idea is to allow users to insert commands into a Quill file to control Quill output (via the Polytext program) to allow multiple column printing incorporating as many of the type styles available on your printer as possible. It should be possible to incorporate graphics but the success of this depends on how wide your printer prints its dots. (!!!) Development has been delayed as the author has been away at college and has been engaged on numerous other projects.

Dennis Briggs has kindly sent me a photocopy of a program by David Stewart in Popular Computing Weekly, 5/5/1988, which gave a similar facility using two keywords: MLIST and SET_PAGE. However this did not allow the incorporation of graphics. If anyone can put us in touch

with the author we could add it to our library.

Miracle Systems have abandoned the idea of a 68020 board and are investigating a 60000 board. (They're cheaper!) Substantial speed increases plus the ability to access more memory are hoped to be the main benefits. Miracle are also reported to have worked out how to allow 32 colours on the QL screen. I haven't the details to hand but I believe that it works by writing to the second screen at 50 Hz (the screen refresh rate.) Each dot then has two colours printed, a different one on each screen, so quickly that it gives a new colour which is a combination of the other two. (Not to be confused with stipples!) I understand that this requires Minerva ROM to work properly.

Andrew Towler of Nottingham has written to say that he is setting up a new QL publishing house, Heirsoft. The first program planned for release is the Heirsoft File Manager, which is a versatile, windowing, multitasking front end for QDOS. Included in the package are a screen dump, file information utility, command file editor and numerous other useful features. Sounds like a simple QRAM style utility, perhaps suited to those who feel that QJump's software is rather intimidating. Also planned from Heirsoft is "Lifestyle", an appointments, diary, address etc program and a text adventure. (this latter to be reviewed in QLLR 2.)

QUANTA have secured the rights to produce the QIMI mouse interface and hopefully production will start soon.

Dilwyn Jones Computing continue to produce good quality and competitively priced software. New progs include Vision Mixer Plus, Cocktails Waiter, Super Disc Labeller, Quick Posters and Home Budget. (All available from either DJC or C.G.H.)

On the hardware front, CL Systems have announced a QL Real Time Digitiser. If anyone has a one please do us a review!

Dr Sohail Bhatti has suggested the formation of a new group to co-ordinate development of a "Super" QL. He should be sending us a circular for inclusion in this issue or next. (He's currently moving house - we'll print his new address when we have it.)

Richard Alexander

TURBOCHARGE - REVIEW

£99.95

By Digital Precision
222 The Avenue, Chingford
London E4 9SE

FOR ALL QLs WITH A SPEED COMPLEX

A few years ago, DP launched a brand new compiler for the QL - Supercharge. Following comments about its failure to compile 100% of all BASIC programs, calls for more speed and even more moans about the copy protection device, TURBO was produced (the special edition of Supercharge fixes most of the earlier criticisms). Turbo received rave reviews in QL World, mainly due to the extra speed which could be achieved over Basic, even when compared with its one and only rival Qliberator. Arguments whether or not Qliberator is a compiler aside, now that Turbo has been around for a while it deserves a further (more in-depth) look.

INVOKING TURBO

Turbo can be made resident in memory or loaded from disk. The former makes for slightly quicker compilation times, but I never choose this option, preferring to keep the QL as empty as possible for testing programs, and besides, it does not make THAT much difference.

Turbo is made up of two programs - Parser_task and Codegen_task. Entering the command CHARGE will load and execute parser_task (1) which allows you to set certain parameters such as the file name, optimise for speed or size, include line numbers etc.. (2). The program will then proceed to turn the current Basic program into a form which can be converted by codegen_task into machine code.

Once the parser has completed its work, codegen_task is then automatically invoked and control is returned to the user who can proceed to do something else whilst the machine code is produced. Parser_task ensures that the user cannot get at Basic and alter the program as it tries to process it.

Compiling itself is quite a quick process: the parsing takes much longer than the code generation itself, but compiling without line numbers and without listing the program helps to speed the job up no end. Still, whilst that is happening, let's look at other areas of the package...

THE TURBO TOOLKIT

(A) THE EXTRA COMMANDS

DP's advertising boasts an extra 200+ commands. However I was a little disappointed to discover that 124 of these are actually Basic functions and procedures which make (innovative) use of existing QL facilities or the new machine code extensions that are provided.

However, I must admit that these are mainly very useful and certainly make full use of the new facilities provided. Also, since the toolkit is only intended for use within compiled programs, it does not really make that much difference. On the plus side, the commands in Basic can easily be modified to suit your needs and you only need to incorporate those which you actually use by merging them with your current program. A configuration program is included to enable you to chop out the desired procedures from within two large files on the supplied disk. Since the two files are just normal saved Basic, this can take a while depending on how many you want and whereabouts in the file that they appear. This could have been made quicker by some form of compression technique - the required commands then being expanded ready for you to merge.

Of course, since the file thus produced is in normal Basic, it can take ages to merge with your own program... most users are probably used to long waits, but the use of a quick loading routine such as QSAVE (Liberation Software) with an added quick merge function would have made the process less painful.

The new version 2.00 of the manual (ask DP for a copy if you have the old manual) documents the use of some of the procedures (by way of example) very well. It also goes on to try and anticipate some of the problems you may face, and how to solve them. The main problem appears to be that you will forget to remove the DEFINE\END DEFINE statements, and so the procedures of course will not work since they are never actually called - solution: remove the offending lines or add a line similar to:

```
100 SB_PROFILE
10000 DEFINE PROCEDURE SB_PROFILE
10010 .....
```

Also, you may need to change the priority of the procedure once compiled (TURBO assumes a priority of 32), using the SET_PRIORITY command, or by specifying the new priority when the procedure is called, ie:

```
EXECUTE flpl_sb_profile_task!10
```

or, add the line (before compiling the procedure):

```
110 SET_PRIORITY 10
```

Some of the more useful additions include 'virtual arrays' which store arrays on a disk or microdrive rather than in memory. These arrays can be up to 3 dimensional (with the routines supplied by way of example) and are then only limited by the amount of space on the storage medium, rather than the amount of memory. Other extensions to arrays allow 'rubber arrays' whereby you can re-dimension arrays (eg. to allow more data to be stored) and yet maintain the original contents!!

eg.

```
1 DIM a(10,20)
2 FOR i=1 to 10
3 FOR j=1 to 20:a(i,j)=RND(0 to 2)
4 END FOR i
```

```
100 DIM a(20,20)
```

You would still have the original 200 values (10 x 20) stored in 'a', but there is now space for another 200!!

Other commands allow you to access the Basic system variables; search or move memory rapidly; list all Basic functions and procedures; stop any task easily by pressing two keys; pipes to communicate between tasks; pass option strings to tasks; WHEN_ERROR (3) on all Qs; error trapping of input devices; full control over keyboard INPUT (e.g. restrict to integer numbers or fixed length strings); FREE_MEMORY (4); set up a new font for the screen, random access file handling; and more...

It is a pity that the run-time toolkit supplied (royalty free) is only available for use from within a compiled program, since it would be useful if a few of the commands such as EXECUTE_A (allowing a task to be stopped at any time by pressing a key) could be used within boot programs.

(B) THE FONT EDITOR

This is quite an easy program to use. You simply call it up and supply it with a name of a font file to begin with (generally one of the many fonts supplied with Turbo). You then enter a character to amend (or the ASCII code of it). A blown-up picture of the

character appears on screen along with a smaller representation (5). You simply move from square to square on the big one using cursor keys (or a joystick) and press space to alter the state of that pixel. As you do this, the smaller representation is altered to show the effect.

Once the character is as you would like it, you simply save the character and move onto another. You can then save the completed font to a disk or microdrive and display the whole font on screen in two character sizes.

Once saved, to use the new font in your Basic program, you merely load the font into memory and then use the new command SET_FONT to assign the new font to as many channels as you like. Simple innit!

Overall this part of the package is quite well designed and easy to use. However it will need amendment to allow full compatibility with MINERVA since the user is only allowed to redefine the character codes 31 to 127.

(C) THE SOUND GENERATOR

I guess this is quite good for the job that it does, but I have never actually found much use for a program of this sort, since it just takes too long to get anything like you desire out of the humble QL speaker (and when you do actually decide on a sound, it is so unrecognisable that it is... {still that's not the fault of the program itself}). The shortcomings of the QL aside, this program can be used to try out lots of different sound effects using the BEEP function. The graphical display on screen is quite good, giving eight 'slider' controls plus a display of the current parameters. You can slide these controls up or down as you wish using the cursor keys, and have the choice of a constant repeat of the sound as you move the them, or a single beep when space is pressed.

This can be a very slow process due to the wide range of parameter values which the BEEP command can take, and so a 'sound juggler' key is included which will slide the controls about at random, hopefully hitting upon something close to your desired sound by chance.

After using the similar sound editor provided with ACT, I missed the ability to play a sequence of different sounds, since this is the only real way of building up some sort of tune or sound effect rather than a simple buzz.

I daresay that some users will find a use for this program, it is given as a little freebie after all!! It can be helpful for experimentation and does after all, prevent having to keep changing the parameters on a Basic line, being unsure just what effect it will have.

Perhaps of greater use is the supplied procedure 'Play_tune', which tries its best (and does quite well!) to make the QL play a tune. You simply provide it with a string containing the notes you wish to play, in normal musical scales, and it then does the job for you. Not too helpful by itself, but once compiled, it can be run in the background to play music whilst your program runs....

SO BACK TO THE COMPILER ITSELF...

Once again DP have outdone themselves with the manual (version 2.00 of the manual is 279 pages long and written on yellow paper so as to make it easier on the eyes). The manual is thankfully relatively straightforward with splashings of humour to break the monotony of such ESSENTIAL reading.

There is a large section devoted to trouble shooting error reports in your programs and this is well written and most useful - it should cover most of your queries over why such and such a program just will not work, and even better, offer simple solutions of how to get around this.

Other sections now include a list of over 60 QL Rom bugs and how to overcome them (if that is possible) both in Basic and under Turbo.

So how to use the compiler?

Once invoked, Turbo displays a menu screen which allows the user to set certain defaults. Navigation around the menu seems positively primitive after using QPAC II, but it is adequate, using the cursor keys to move from box to box and space to alter options (note that the initial value of most of the options can be set by a configuration program, meaning that you could simply compile some programs by merely pressing ENTER!!). The available options are (some of which may not be accessible if they are irrelevant in view of your other choices):

1) Set the data space for the compiled program.

This determines how much space is set aside for the storage of variables and workspace within the task. It is generally best to opt for a relatively high figure for programs which use a lot of strings (eg 20K) and then reduce this using a supplied utility once the program has been compiled. Using this method means that you will not be constantly bombarded with the error 'Out of memory - increase dataspace' when you try to EXEC the final program. Besides, since data space usage within a program alters as the program is being used, it is always better to have slightly too much set aside just in case.

Once you find a figure which the program seems to run quite happily under, setting it in the first line of your program with 'DATA_AREA' prevents mistakes during later development when you forget how much you used! One of the supplied toolkit utility programs can also be used to monitor usage of dataspace within a compiled program so that you can see what would be an optimum value.

2) Set the buffer space for TURBO itself

This determines how much room there is for Turbo to operate in memory. It will affect the amount of compilation time spent accessing your files, so increasing this figure as your memory permits will decrease compilation time (I find that a setting of 50K is more than adequate for even the biggest programs).

3) Program size <64K

Setting this at <64K will ensure a more compact final program. Although Turbo will report if your program is too long to use this setting, it cannot itself alter the setting because it would mean re-starting compilation rather than merely altering one or two things in the final program. The reason for this is that this setting determines whether or not the machine code produced uses 16 or 32 bits for addresses. It would however be nice to be able to set this at the start of your program (eg with a Rem statement) so as to prevent a lot of aggravation when you later forget to set this! You do eventually begin to recognise when a program is going to exceed 64K, but it is always worth a try compiling without line numbers and ensuring all of the windows used are opened by the program itself, since this may reduce it below 64K.

4) Include Line Numbers

Unlike Qliberator which uses this option to allow a table to be set up for computed RESTORES, GOSUBs and the like, Turbo will not handle these and so this option is generally only included for debugging your program. Without this option enabled, any errors when your program is run will be reported as occurring at line 0 (not very helpful is it?)

You can also select whether or not Turbo is to display the line numbers as it compiles its program. Turning the display off allows Turbo to run at its highest speed, but it means that there is hardly any indication on screen that Turbo is still working!

5) Optimisation (Brief, Fast, REMs)

This option allows the user to decide just how much to trade-off speed against memory usage in the final program. Even selecting 'Brief' gives a massive increase in speed, still much quicker than Qliberated programs. 'Fast' is only useful in programs which do large numbers of calculations since it cannot increase the speed at which QDOS operates (you need Lightning and/or Minerva for that)! However it must also be borne in mind that the manual does contain many useful hints on obtaining the best speed from your program in any case.

In most programs which use both calculations and writing to the screen, the user is best to use 'REMs', which enables you to pin point specific areas of your program which should be compiled to give optimum speed, allowing the rest to be compiled with memory space in mind. But how to find these areas?? Thankfully there is a supplied toolkit program which will display the amount of times each line in a program is called, enabling you to discover which areas should be improved by this method.

Using either FAST or REMs will increase the length of your compiled program quite drastically and so the user must bear this in mind. Unfortunately, despite 128K on the standard QL (I remember the 1 - 16K of the ZX81!), memory is generally at a premium. I generally find that the compiled program uses more memory than the Superbasic program, although this can be alleviated to a certain extent.

6) Open windows

This option will make Turbo include code at the start of the compiled program to open windows so that it begins in the same state as it would in Basic. To do this, it copies across the current characteristics of the specified windows which are already open (generally #0 to #2). However, this involves some unnecessary code and it is generally better for you to open ALL of the channels used by your program. I just generally include at the start of my programs the line :

```
10 OPEN #0,con_448x40a32x216:OPEN #1,con_448x200a32x16
```

(WARNING: you have to be careful now because if you run the program in Basic with this line intact, you will find that the new #0 will not accept many direct commands, so remember to put a REMark before the start of the line if you want to run it in Basic).

NOTE that Turbo'd programs can have a maximum of 16 channels open at any time. If you use a channel number above #15, Turbo will report an error on compilation. This is not too great a restriction since there will be very few programs which need more than 16 channels open at any one time. Besides, ensuring that channels use consecutive numbers helps to save memory on the QL.

7) Sound on/off

This turns the sound on or off during compilation. In fact it does not make all that much difference since there are only a few beeps, as you move around the menu, when compilation starts and if there is an error.

8) List program

This prints out the current program as it is compiled. It therefore slows compilation time considerably.

9) Task name

What will you call your finished program? This is the name that will appear on any list of current jobs using JOBS or LIST_TASKS.

10) File name

Where should the compiled program be stored? Note, you should ensure that the medium used has some free room since Turbo creates extra work files during compilation.

11) Freeform/Structured

This again alters the length of your compiled program. A structured program is one which has all of its procedures and functions at the end of the main program - all well written Basic should be like this since it is otherwise harder to follow the flow of a program.

12) Create strings

All strings within a compiled program must be DIMmed to set their overall length. Generally a string used within a program will automatically be set to 100 characters by Turbo unless it has been told otherwise. Setting the correct length of the strings will save memory and also prevent only part of what was intended being printed out on screen. However Turbo cannot know what to do with strings which only appear within a procedure or function - is it that they are only used within that procedure or should they have been declared at the start of the program??? This option allows the user to tell Turbo what to do about such strings. Turbo can just ignore them and tell you about them during compilation, or make them into global variables for the whole program.

COMPATIBILITY

Unfortunately no compiler can be said to be 100% compatible with Superbasic. Of course there are some meaningless commands which Turbo will ignore (LIST,SAVE,LOAD etc) which have no meaning within a machine code program. However there are certain people who argue that for compatibility, buy Qliberator, and so this topic must raise its ugly head.

To help you in your struggles, Turbo does include an auto-corrector, which will try to make sense of an unwieldy program. It actually works quite well, but of course it can only make educated guesses about what the program is attempting to do, and so it is always best to trim your own programs by hand. It will of course warn you about any things which it does alter, so that you can prevent a program jumping out of a procedure with GO TO for instance, or ensure that END IF appears in the correct place. I have only come across one program which completely flummoxed Turbo, but then it did have references to NEXT loop several hundred lines after the END FOR loop!! (No names mentioned, but the program never did run 100% in Basic anyway before I got my grubby little paws on it!)

To be honest, if you are writing your own programs there is no real problem, since you soon get used to Turbo's idiosyncracies and begin to write Basic with them in mind. However, other people's programs may cause problems. The main difficulty appears to lie in the fact that Turbo will not compile computed RESTORES or DATA statements. The latter is understandable since by its very nature, DATA should be fixed amounts and not be made up by reference to variables. However it would have been nice for Turbo to be a little more forgiving here and compile the DATA if there is no fear of the values changing later.

For example, Turbo will not compile DATA 5,a+6,7,BI*3; but then it will not compile DATA 2+2076,3,4*6 either!! (The latter could be used to make programs more readable and therefore easier to debug).

As for computed RESTORES, eg RESTORE 1000+a, this is widely used in adventures and would mean a lot of extra IF...THEN.... statements. Surely if the program were compiled with the option to include line numbers, Turbo could allow this since the data tables required must exist. If it is a question of wasting space in programs which do not use such commands, then surely a software switch (ie a REM statement) for the user to specify if he wants this option could be incorporated.

Things outside the programmer's control:-

1) QPAC II/QRAM

Programs compiled by Turbo cannot be used under the QPAC system unless the 'impure' option is used for different reasons. However, this has no real effect except that more memory is needed to run the program than would otherwise be the case. There are other problems in that when a compiled program generates an error, it will not wait for a key to be pressed before it returns to Basic, so unless the unlock option is used from QPAC, the user will miss the error reports due to QPAC's screen refresh. Whilst I'm on the subject, `PARSER_TASK` should be invoked with the unlock option on since you will lose any error reports during compilation and the screen will look odd when `CODEGEN_TASK` is started up (despite the Turbo configuration program asks if Turbo will be run within a QRAM environment). This means that unless you have the original QRAM 'unlock' program, you will be unable to initiate Turbo using 'CHARGE', and would need to use QPAC's 'EXEC flpl_parser_task,i,u:EXEC flpl_codegen_task,i,u'

2) QPTR system

Turbo cannot compile programs which use these machine code extensions (not to worry, Qliberator can) since according to DP, they include procedures which (illegally?) alter variables used by Basic directly. - Arguably though, since Basic allows this, why not Turbo??

3) Minerva (Turbo is reportedly being updated)

There are several problems here.

First of all, when I received the latest Minerva Rom (v1.80) I discovered that not all of the commands in the turbo toolkit were being linked into basic. This was because of two of the names (in v2.10 of the toolkit) giving errors when being linked into Basic which Minerva recognised and stopped. This was easily remedied using an editor and indeed has now been fixed by DP.

Now the main problem seems to be to do with the integer tokenisation used by the latest Minerva to speed up Basic programs. Turbo will not compile Basic programs unless the integer tokenisation has been turned off by `POKE \\212,128` and the Basic program saved and then reloaded.

Another small problem is that working under Minerva, CHARGE will no longer pass a file name to the `PARSER_TASK`. Instead the default file name is given as 'PARSER_TASK'!! On top of this `CODEGEN_TASK` is no longer started up automatically and therefore must be started by 'EXEC flpl_codegen_task' after parsing. I find it easier now to compile programs by using the line:

```
EXECUTE_A flpl_parser_task:EXECUTE flpl_codegen_task
```

These are however minor problems and may even be fixed by now. They do not affect the final programs, and in all Turbo and Minerva compliment each other in achieving maximum speed with very little effort from programs. In fact, the minor limitations Turbo imposes, does help with ensuring that Turbo'd programs will run on ALL QL's, since many of the extra features given with Minerva are also available from within the Turbo toolkit. For instance, Turbo may not be able to compile programs written on Minerva using string selects; but Turbo itself allows a means of doing exactly the same (except in a method that can be used on any Basic program on ANY QL).

OVERALL

Turbo is now beginning to show its age and is in need of a little amendment to bring it up to date to cope with the latest developments in the QL world. It is hard to keep up with a rapidly changing product such as Minerva, but Liberation software (authors of Qliberator) appear to manage. Hopefully once Minerva development settles down a little more, older software can begin to catch up.

I have now received a preview version of a later version of Turbo, and although it still is not fully Minerva compatible, the toolkit seems to suggest that it soon could be - for instance there is now a function to return the base of the system variables. Other new words hinted at in the toolkit seem to suggest that Turbo may also be being altered so that many of the front panel options can be set at the start of a program - more news if I receive any.

The speed improvement by Turbo is excellent although the code produced is still not as compact as it could be. If anyone has any hints on how to reduce the memory requirements for compiled programs then please write in to QLTR. It is a shame that the manual fails to give this as much attention as it gives to increasing speed. The only hints I have (except as included above) are:

- 1) Use BRIEF, STRUCTURED and Line Numbers OFF
- 2) Do not include unnecessary REMARK statements (except to turn on and off speed optimisation if wanted) since Turbo includes these in the final program.
- 3) Delete any excess spaces in the program.
- 4) If a procedure is only called once, consider incorporating it into the main body of the program rather than as a separate procedure. (It doesn't help readability, but then...)
- 5) Open all windows used by the program from inside the program, rather than rely on Turbo to do it for you.
- 6) If the auto corrector issues you a warning, try to fix the problem by hand, since this undoubtedly saves space (and ensures the program runs as you want it to).
- 7) Compile the program with only those extensions in memory which the program uses.
- 8) Use floating-point flags instead of integers.

A review such as this (long as it may be!) can only hope to touch upon the extra facilities given by Turbo. It may seem a little too easy to criticise the program, but I am actually still pleasantly surprised by the new things I discover each time I use it. I bought the program some three years ago, and have never regretted it since. Admittedly the program is unlikely to drastically alter your life (unless the manual gives you a hernia), but it certainly will breathe new life into your QL and alter the way in which you program the computer. No longer will you consider throwing away those old programs which were too slow to be of any real use except as a programming foray. The only problem now is that I find I hardly ever write Basic programs to run in Basic - oh well, I guess I'll just have to put up with all that extra speed.

Rich Mellor

NOTES

(1) The command CHARGE is supposed to be able to take one parameter specifying the destination file for the compiled program. This has never properly worked unfortunately (it is odd that DP have never seen fit to fix this), since the specified file name is not necessarily the one offered by the front end as the default.

For example:

```
CHARGE flp2_GAME_exe
```

will be displayed by the front end as:

```
flp1_flp2_game_e (it the file already exists)
```

or flp2_game_exe_ (if it does!! - not what you would expect!)

(2) It would have been nice to be able to set many of these defaults from REMark lines at the start of a program so that you did not have to remember to change them every time that you compiled a developing program - it is all too easy to forget to set the code size to >64K or to specify 3 windows to open. You can set the data space to be used by the finished program, but that's all unfortunately.

(3) After my comments and plea for help regarding WHEN_ERROR in QLTR 2, Simon Goodwin has contacted me and informed me that I must use a RETRY_HERE command somewhere in the program to direct where the compiled program is to try to recover from the error. However, further to this, I have discovered that Turbo suffers the same limitation as Qliberator in that errors which are seen as programming errors (e.g. LET vars='VT') will not be error trapped in the compiled program, even if you use the normal WHEN_ERROR provided by JS, MG and Minerva Roms. Therefore the following will work when compiled:

```
100 DATA_AREA 5
120 WHEN_ERROR 1
130 PAUSE:PRINT 'error - line';ERLIN%
150 PAUSE
160 a=0:b=b+1:RETRY
170 END_WHEN
180 PRINT 'hello'
190 a=0:b=2
200 RETRY_HERE
210 REPEAT loop
220 AT b,a:PRINT 'HI'
230 a=a+1
240 END REPEAT loop
```

However, changing lint 190 to a='x':b=2 will stop the compiled program with an error because it is a programming error.

(4) There is a little oddity with the command 'FREE_MEMORY' in that whilst from Basic it gives the amount of memory unused by the QL, from within a compiled program, it only gives the amount of free data space of that task. Not to worry! There is a supplied function which can be used to discover the amount of free memory outside of the task if need be.

(5) It would be nice to be able to set the character size of the smaller representation, since on some Qs (pre JS Roms), altering certain columns (ie columns 0,6 and 7) of the characters has different effects depending on the character size.

MORE BITS AND PIECES

MICRODRIVE ERASURE

Dennis Briggs tells me that, contrary to certain advice printed earlier in QLTR, it is not possible to erase mdvs with a magnet or with an ordinary bulk tape eraser. To erase a tape it must be subjected to a strong high frequency magnetic field. (Info on this from Ablex and Maxell.) I usually find that putting an important file on a mdv is quite enough for the mdv to lose whatever is on there!

In an earlier issue we printed a short item that suggested using the pseudo-random number system variable SV.RAND (word) as a counter that is incremented 500 times a second. Ian Bruntlett points out that this may be of use if the software does not rely on random numbers. However it is machine specific and varies from ROM to ROM and may not get updated as frequently if there are quite a few jobs on the machine.

If you are using Minerva using then PRINT PEEK_W (!!46) will read SV.RAND.

MACHINE CODE UNDER QDOS

This will show how to add a new function to the machine code commands available from Superbasic. This doesn't use any particular tricks and uses a minimum of mathematics operations.

The function which will be linked into Basic is the well known function 'SGN'. SGN is defined as $X / ABS(x)$, with '0' as an exception. Therefore the following results are given:

```
PRINT SGN(0)  ---  0
PRINT SGN(-3) --- -1
PRINT SGN(3)  ---  1
```

Simple eh?

First off, we need to define a few QDOS constants to make it easier to read the program listing. For example RI.DIV is a mathematics constant which is used to call the division routine, and is made equal to \$10. Instead of using '=' as in Basic, you use 'EQU' (for equals) in assembly language.

```
BP.INIT  equ    $110
CA.GTFP  equ    $114
RI.EXEC  equ    $11c

RI.DUP   equ    $16
RI.ABS   equ    $12
RI.DIV   equ    $10

ERR.BP   equ    $ffffffl
```

A few comments on the names:

BP.INIT is a call to a Rom procedure which initialises new Basic procedures and functions.

CA.GTFP fetches a floating point variable given after a function name eg SGN(100) or SGN(x) - CA.GTFP allows the machine code to read the value (100 or x)

RI.EXEC carries out a maths operation held in d0. The other three RI.'s are the mathematical operations themselves.

ERR.BP is an error code.

After the constants have been defined, the new procedure/function must be linked into Basic.

```
lea.l    proc_tab,a1
move.w   BP.INIT,a2
jmp      (a2)
```

Proc_tab is the start of a table containing the new names of the functions/procedures. The first two words are zero to say there are no new procedures and the end of the procedure definition has been reached. The next word is 1 to show that there is one new function. The next word is dealt with later. Then comes some bytes which give the length of the new name, followed by the name of the function (SGN). The last zero word shows that the end of the function definitions has been reached.

```
proc_tab
dc.w     0
dc.w     0
dc.w     1
dc.w     func_sgn-*
```

```
dc.b    3,'SGN'
dc.w    0
```

So far all is clear. Now a brief word about passing parameters. When a machine code procedure or a function is called, the interpreter creates a list of all of the parameters which are included. A3 points to the start of this list and A5 to the end. When A3 and A5 hold the same value, no parameters have been passed. Each entry is eight bytes long, therefore $(A5-A3)/8$ gives the number of parameters. We need not trouble too much about this list since we only need one parameter. QDOS sorts all that out for us, we must just make sure that until these parameters are actually fetched by our routine, that A3 and A5 are not destroyed.

Now we come to the first label:

```
func_sgn
    move.w    CA.GTFP,a2
    jsr      (a2)
```

The word in the table which we didn't comment upon tells the routine, when it is called from Basic that func_sgn is the start of the actual machine code to carry out the task we wish SGN to do. Next we need to call the QDOS routine to fetch the parameter passed to our function. Basic can normally cope with strings, floating points and integers. Machine code can also cope with long integers (32 bits), whereas Basic lacks this capacity. One generally wishes to pass more than a word (32767), so we must assume the parameter is in floating point format. Such a number takes six bytes to store it in memory.

To fetch the floating point, we use CA.GTFP which is a vector, ie. the address of the actual machine code routine to call is only stored at CA.GTFP. We must therefore store this in a2 and jump to the subroutine pointed to by a2. We do not use JMP since this would not return us to our own machine code. This routine will fetch the parameter with D0=0 to signify no error. We cannot be sure that a parameter has been supplied and so must therefore check to see how many have been given. The number of floating-point parameters supplied is returned in D3. This routine will also convert numeric strings into floating point etc, so that SGN('12') and SGN(12) are the same.

In this routine, we only need one floating point parameter, and so we must check to see that this is the case. So to check, we will subtract 1 from the value in D3 and see if it is 0 or not. When a data register is set to zero, the zero-flag is set. There are two ways to see if the zero flag has been set: EQ (the number is 0), NE (none zero). If the number of parameters is not 1 then we leave the routine with a 'BAD PARAMETER' error, by jumping to error_bp in the program.

```
subq.w    #1,d3
bne.s     error_bp
```

When we reach here, there must only be one parameter. Now we must check to see if the parameter is 0 since a division by 0 will give an error. Now it's time to clarify the principle of the arithmetic stack. This works as per a normal stack ie, each new word is stored below the previous one. Each entry in this stack is a floating point (ie 6 bytes). A1 points to the top of this stack. A6 points to the basic variables and can move around in memory, therefore the stack is relative to A6. When CA.GTFP is called, it points A1 to the position of the stack relative to A6 so A6,A1.L points to the first byte of the six bytes of the first entry. So why is the '.L' placed after the A1? This ensures that the whole of the longword held within A1 is used as the pointer to the stack.

Now to see if the number is zero.

So how do we test if all of six bytes are zero (The maximum number of bytes a register can hold is 4 - a longword)?

Well, we must first place the first four bytes (ie a longword) into a data register. If they are zero, the zero flag will be set. The remaining two bytes must be 'or'd with the first four, and only if these are all also zero will the zero flag be set.

```
move.l    0(a6,a1.L),d1
or.w     4(a6,a1.L),d1
beq.s    sgn_end
```

Now if the zero flag is set (EQ), the program will jump to the end routine. Zero equals zero!

```
move.w    RI.EXEC,a2
```

This gives the vector to call the arithmetic routines. This vector is handled just the same as any other. We need to use it to call three different things to do either $x/ABS(x)$ or $ABS(x)/x$.

First of all we need to duplicate the entry on the stack so that we have two numbers to divide upon the stack. Next we need to ABS one of those entries, and lastly divide the two entries.

To do this, we place the number of the function in D0 before we jump to the vector subroutine. We can therefore create a list of these functions since each answer will be placed upon the stack in place of the different entries.

To duplicate the first entry on the stack, load the function (RI.DUP) into D0 and call the vector (A2):

```
moveq     #RI.DUP,d0
jsr      (a2)
```

Now to get rid of the sign on the second entry:

```
moveq     #RI.ABS,d0
jsr      (a2)
```

Now, finally to divide the two entries now on the stack:

```
moveq     #RI.DIV,d0
jsr      (a2)
```

This leaves just one entry on the stack - the result of the division.

Now at last we reach `sgn_end`. Of course the program will have jumped here if the parameter had been zero. There is no need to alter the stack since it already contains either zero or the result of the division (-1 or 1). We have not altered D0 and can be sure that CA.GTFP will have set D0 to 0, so we can safely return to Basic without any error.

All that is left to do is to tell Basic which type of parameter is being returned by the function. This is signalled by D4. This can have one of three values (Strings=1, Floating Point=2, Integer=3). In this case we therefore need 2. We can then return to Basic:

```
sgn_end
moveq     #2,d4
rts
```

Finally we need the code to signify a BAD PARAMETER error, by loading this into D0 and returning to Basic, where the error will be reported in #0.

```
error_bp
```

```

moveq    #ERR.BP,d0
rts

end

```

So, the finished program will look like:

```

BP.INIT  equ    $110
CA.GTFP  equ    $114
RI.EXEC  equ    $11C

RI.DUP   equ    $16
RI.ABS   equ    $12
RI.DIV   equ    $10

ERR.BP   equ    $ffffff1

        lea.l   proc_tab,a1
        move.w  bp.init,a2
        jmp    (a2)
proc_tab
        dc.w    0
        dc.w    0
        dc.w    1
        dc.w    func_sgn-*
        dc.b    3,'SGN'
        dc.w    0
func_sgn
        move.w  CA.GTFP,a2
        jsr    (a2)
        subq.w  #1,d3
        bne.s  error_bp
        move.l  0(a6,a1.1),d1
        or.w   4(a6,a1.1),d1
        beq.s  sgn_end
        move.w  RI.EXEC,a2
        moveq   #RI.DUP,d0
        jsr    (a2)
        moveq   #RI.ABS,d0
        jsr    (a2)
        moveq   #RI.DIV,d0
        jsr    (a2)
sgn_end
        moveq   #2,d4
        rts
error_bp
        moveq   #ERR.BP,d0
        rts

end

```

(NOTE the original article was much longer dealing with fetching more than one parameter off the stack, but this has been omitted due to wanting to keep it simple, and some terrible German words to translate!!)

Translated from the German original in QUASAR 21 by Rich Mellor (apologies for any artistic licence employed to overcome my lack of German language!!)

Note that we have two further issues of QUASAR which people are welcome to borrow to translate more articles. Our thanks to Rich for his work on this.

DILWYN JONES COMPUTING PROGS

41 Bro Emrys, Tal-y-Bont, Bangor, Gwynedd LL57 3YT
TEL: 0248-354023

This company has been set up by the author of Page Designer 2 to sell some new, more specialised utility software. All Dilwyn's software is very robustly written, TURBO'd and happily multitasks along with other programs (all have an in-built screen refresh key in case of corruption by other programs). All of them are supplied with extensive manuals in the form of a Quill document or a neatly photocopied manual. At the time of writing, due to the shortage of microdrives, if you want the software on microdrive, Dilwyn asks that you supply your own media. The software is available on 3.5" disk, 5.25" disk (80 or 40 track), or microdrive (Vision Mixer requires two microdrives). It is easy to convert the programs to run from any drive if necessary - details are given in the manuals.

BASIC REPORTER (128K) £10

This program provides the Basic programmer with lots of useful different utilities. The main part of the program is a TURBO compiled task, but there are also some supplementary files. First of all let's look at the compiled task.

On loading the program, the user is presented with a menu giving a wide range of different functions. Many of these things are available from things like Liberation software's QREF, but this program is much more flexible and gives much more information about the Basic program currently in memory. The menu itself is easily used, since you can either use the cursor keys and space to move the selection bar onto the desired option, or press the letter given before the option.

The options include details on different variables used by your program (and machine code extensions which are available for use if so desired!) and also a Basic trace utility which works even with Minerva. The trace function itself is not as complex as the utility Supertrace by Stack Software, but provides a small window in the top right of your screen giving details of the current line number and the number of the command on that line being executed.

You can get lists of all machine code extensions available, or only those used by the program, arrays used by the program, Basic procedures or functions used by the program or merely all of the variables which the program refers to. The output (which can be directed to a printer by the way) will list the items and tell you what type they are (eg strings, integers or loops). You can choose to specify a range of line numbers that the program is to operate in, or get a report on the whole Basic program.

Once you have obtained a list of all of the variables (etc.) used by the program, you can then use reporter to find out which lines they occur on; and where procedures or functions (Basic or machine code) start and finish (either by use of a return or end define) and where they are called from. When you wish to do such a search, you are asked to input the name of the variable, procedure or function to look for. You can however opt to enter merely part of the name and get a report of all the variables which begin with 'score' for example. The output can be sorted alphabetically or by line number if you prefer; but either option may take time.

One thing that Reporter allows you to do which QREF cannot is to report where keywords like 'SElect', 'REMark', 'GO SUB', 'GO TO', 'REMAINDER', and 'WHEN'. Here the problem is that they do not appear in the name table and so QREF cannot see them. Reporter must physically search the program to find these!!

Other useful things provided by this part of Reporter are a report on the current usage of memory, and a list of all unset names which can be used to identify mis-typed variables.

This part of the package is indeed a very useful debugging aid, and with space to handle about 2,300 entries in the name table (the program can be altered to handle more if necessary) should be able to give useful reports on most programs. It will undoubtedly be of aid to Basic programmers who wish to squeeze their programs into as little memory as possible, by identifying variables and procedures which are little used.

Also provided in the package is a Basic program which will indent your Basic listing so that it is more readable. It is not compiled and therefore suffers from the problem that on pre JS Roms, it cannot handle lines of more than 127 characters. However it is still useful, especially for magazine listings. Also provided are four machine code extensions: three of which allow you to convert text to upper or lower case; whilst the other allows you to change names of variables quickly and easily.

VISION MIXER (256K) £10

This is a program which will take up to 25 normal screens and display them on a monitor or TV set in a new and exciting way. Instead of simply loading them one after another, this program allows the user to load in a series of screens and then use over a hundred different video effects to switch between the screens. This may be useful for any sort of eye-catching display such as in a shop window for advertising, or for lectures.

All of the screens must be in the normal 32K format since this increases the speed at which the program can work. It is a pity that the program cannot load and decompress any pictures stored on disk since this would increase the number of pictures that can be stored on a disk (or heaven forbid on a microdrive!). Another short-coming (?) is the program's inability to show both mode 8 and mode 4 pictures in one sequence - it would therefore be useful if Dilwyn were to supply a program which could convert a mode 4 picture into mode 8 without any flashing pixels etc which this normally causes. It is an understandable limitation since it would otherwise affect the switching of the screens if the mode had to be changed.

The 7500 word manual is highly informative and leads you through loading a demonstration set of screens to developing your very own rolling credits to film and put on the end of a video tape! There is however one small typing error in the manual in that it tells you to COPY flp1_demo_seq TO flp2_demo_snl, when it should be COPY flp1_demo_seq TO flp2_demo_seq. Still, with the length of the manual in mind, such a minor mistake (although frustrating as it may be to new users) can be forgiven.

When the program loads, it asks you to enter the mode (4 or 8) in which the pictures are to be displayed, and the colour of a blank screen used by the program for certain effects. You must then enter a list of screens which the program is to use. Once such a list is created, it can be saved onto a file for editing later or for use by the program at a later stage to save you the trouble of entering all of the names again. It is odd that you have to enter the number of pictures before their filenames, since surely the program could count them itself, if you had to press a key to end the list of filenames.

Once this is done, you must select the effects (or mixes) to be used by the program to switch between the screens. You can specify how many times a chosen sequence is to be run, either infinitely or up to 32,000 times! You then have the choice of using a random selection of the video effects provided by the program (excluding chosen mixes) or creating your own sequence. It is relatively simple to create a sequence of effects, since it merely involves moving a bar up and down the effects, and choosing one effect for each picture. A facility is included to enable you to see the effect in action before you decide so that with a little time and effort you can produce quite spectacular results. Not to worry; once again you can save a specified sequence onto a file for use later.

The program will multitask, but you are told to turn on the cursor if you wish to do so to avoid being locked out of the program at a future date. Unfortunately, leaving the cursor on will disrupt some of the video effects, namely any which involve panning or

scrolling the screen. There are only a few of this sort of effect and so they can easily be avoided.

In all the program is extremely useful for creating interesting displays and is really quite simplistic to use after a little practice. The switching between pictures is extremely smooth and very quick (depending upon the effect itself!) and adds that little more to your different advertisements.

WORDSCHECK (128K) £6

This program is used to produce a report on the words used within any document (a normal text file, or a Quill_doc). On loading it asks for the name of a document to check. If you delete the supplied file name and enter a question mark, the program will produce a directory of the desired medium - wild cards are implemented as per toolkit II so, that typing 'flpl_doc' will produce a list of all quill documents on flpl_.

The program can cope with any drive name, and will proceed to load the document into memory if there is enough available memory (else it will work with the document on disk). Next it proceeds to count the total number of words in the document and also to count the number of different words along with the number of times that a word appears. This can take quite a while if the document is quite long, since it can take quite a while to compare each word with the list of previously used words.

Once this is done, you are asked if you want to set limits on the word lengths and number of occurrences which will be listed. You can then also sort the words alphabetically, which uses quite a quick (but effective) sort routine. The next option is to print out the word list on screen or to a printer.

The output of the program is a list of the words used in two columns. Each word is followed by the number of times which it appeared in the document, in brackets. This can show some odd things about your word usage, such as in one review 867 words long, I had used 356 different words, in which I had used 'can' once, but 'cannot' 33 times!! - Odd, considering I'd always considered it to be a very positive review!. After the list of words, again the total word count and number of different words used is printed.

This list can also be saved onto a file on disk, and then accessed from Basic. A method to do this is given in the manual. It must be noted that the program does have its limits. It will truncate any words over 20 letters (no real hardship I'm sure), and can cope with documents up to 32,000 words in length with 1,000 different words in it. Believe me, these are not really limitations upon the normal use of the program.

So, what can the program be used for??

Well I daresay that many different statisticians would find the program of use, but it can also be useful for example if you want to compress a file (eg the locations in an adventure), since you could see which words you would be best to compress. Other uses no doubt exist, possibly even for encoding/decoding messages.

In all the program is well written and does its job well. I thought it a pity however that you could not opt to make the program compare words with the case independent since 'Can' and 'can' are given as two different words by the program.

Rich Mellor

EDITORIAL INTERRUPTION:

I spoke briefly to Dilwyn Jones regarding WordsCheck and he informed me, in response to my passing on some criticisms of a review copy we had received, that there was, indeed, a bug in early versions of the software which meant that the table which held the words and the one which held the count variable, got out of step with each other. A free upgrade can be had from Dilwyn if your version shows signs of errors.

SUPER TOOLKIT II

by Stephen Bedford

Part 2

5 File Maintenance

TKII improves upon the standard set of procedures in two ways. First, the existing commands COPY, DELETE and DIR all use the default directories. The second area of improvement is the addition of several new commands which implement two new types of operation which significantly ease file handling on the QL: wild card operations and overwrite operations.

5.1 Wild Card Names

In the TKII manual it explains that wild card characters are not used, instead any missing section of a name is treated as a wild card. However, in the first part of these notes I explained that an underscore is a wild card character. In fact substantially these two views are the same but I believe that considering the underscore to be a wild card character is easier to understand. The use of wild cards in this section is the same as used for defaults directories explained in the first part (section 4.2). However for the notes to be correct a stricter definition of what the underscore can represent must be defined:

An underscore can represent a null string or any series of characters that do not start with a delimiting underscore and end with either an underscore or the end of the name.

This is consistent in many cases with saying that the missing section is treated as a wild card.

The following example explains the definition. Suppose we have a disk in flpl_ with the following files:

```
basic_mandelbrot_bas
basic_games_bas
basic__jobs_bas
basicprogram_bas
```

If the data default is set to flpl_ (DATA_USE flpl_) then

```
DIR basic_bas
```

would show the following

```
basicprogram_bas
```

The underscore is representing program_. The other files do not match since the underscore following basic is a delimiting character and the wild card cannot represent a string that starts with an underscore. However,

```
DIR basic__bas
```

Would show all the files. For the first three file names the first underscore in the wild card name is the delimiting character and the second represents mandelbrot_, games_ and jobs_ respectively. For the last file name the first underscore is set to a null string and the second matches program_ as before.

Thus, a wild card name of flpl_basic__bas could match a file name of flpl_basic_mandelbrot_bas and it may be considered that either mandelbrot is the

missing section of the filename or that the second underscore in the wild card name matches `mandelbrot` in the file name.

However, if a disk contains a file with the name `letter` on it then the command `DIR l_` will result in the file `letter` being listed. Yet an underscore does not appear in the filename suggesting that the underscore is a wild card matching a series of characters ending with the end of the file name. In section 4.2 of the TKII manual it explains that if a default directory is set that does not end with an underscore then an underscore is automatically appended. This may be considered to be the case for wild card name too. Thus in this example `DIR l` would also result in the file name `letter` being displayed.

It doesn't really matter how wild cards are defined. The important thing is to realise that they are very useful. Practice in the use of wild card names will hopefully bring understanding.

Some of the examples in section 5.1 of the TKII manual will not work. It could be that they are misprinted and some of the underscores should in fact be two underscores. In the last example a wild card name of `flpl_old_list` would not match the file names `flpl_old_jo_list` and `flpl_old_fred_list`. A wild card name of `flpl_old__list` would be needed.

5.2 Directory Listing

As well as the standard `DIR` command TKII also makes available `WDIR` and `WSTAT`. All of these use the default data directory and may be passed wild card names. Also the output of the commands may be redirected using implicit channels as shown in part 1.

If we have a disk in drive one with a name `TKII Notes` and the following files on it:

```
TKIIa_doc
TKIIb_doc
TKIIjob_doc
```

Then `DIR flpl_` would result in the following display:

```
TKII Notes
1347/1440 sectors
TKIIa_doc
TKIIb_doc
TKIIjob_doc
```

`WDIR flpl_` would give:

```
TKIIa_doc
TKIIb_doc
TKIIjob_doc
```

and `WSTAT flpl_` would give:

```
TKIIa_doc
 16590 1990 Jul 06 20:25:38
TKIIb_doc
 17065 1990 Jul 10 15:53:29
TKIIjob_doc
  8412 1990 Jun 23 17:16:57
```

Note the amount of space on the disk is shown in sectors (blocks of 512 bytes). The file sizes are shown in bytes. However, the space for a file is allocated in groups of three sectors thus `TKIIa_doc` would use 33 sectors, `TKIIb_doc` would use 36 and `TKIIjobs_doc` would use 18 sectors, that is 63 sectors in total. The other 6 sectors that have been used are for the directory and map (a directory of a blank disk will

show 1434/1440 sectors).

WSTAT is very slow on microdrive.

5.3 Drive Statistics

The command `STAT`, shows just the name and space available on a disk. In the above example the display would be:

```
TKII Notes
1368/1440 sectors
```

To get full information on a disks contents type:

```
STAT flpl_ : WSTAT flpl_
```

Note in the contents section of the TKII manual a command `ASTAT` is mentioned. This command, which should produce an alphabetic list of files, is not described elsewhere in the manual and in fact is not implemented in the versions of TKII that I own (versions 2.12 and 2.13).

5.4 File Deletion

The `DELETE` command has been modified to use the data default directory. Thus, for a machine with floppy disks attached just after booting, the command

```
DELETE boot
```

would delete your boot file contained on `flpl_`. For a microdrive only system the same command would try to delete a file named `boot` on `mdv2_`.

A new command has been introduced, `WDEL`. This command will accept wild card names as a parameter.

Suppose we are using the same disk as above ie containing the files:

```
TKIIa_doc
TKIIb_doc
TKIIjob_doc
```

Typing the command

```
DELETE TKII
```

would result in disk would spin and no error message would be produced yet nothing would be done: the file `TKII` does not exist. Whereas the command

```
WDEL TKII
```

would produce the following response

```
FLP1_TKIIa_doc..Y/N/A/Q?
```

meaning, is this file to be deleted (Yes or No), are All files that fit this wild card to be deleted or is the operation to be Quitted. So, to delete all except the first of the files that fits the wild card, first respond with N then A.

It is suggested that the option of deleting all matching files is not used until familiar with this command and wild cards. On a machine running MS-DOS a prompt similar to the one described above is not given `DEL TKII` would go ahead and delete all files that match!

5.5 File Copying

The standard COPY command has been modified to use the DATA and DEST default directories. Thus, the command

```
COPY boot
```

will copy flpl_boot to ser1. That is, assuming the defaults have not been altered. So, if a printer is attached, the file will be printed.

A further alteration to the COPY command is that if the destination file already exists permission to overwrite is asked for. Thus, typing

```
COPY tkiaa_doc TO tkiib_doc
```

the following prompt would result

```
FLPl_tkiib_doc exists, OK to overwrite..Y or N?
```

Very much like the QUILL Save operation.

The COPY command has also become more "intelligent". The file header is automatically either copied (eg making a copy of an executable file) or not (printing a file) depending on the file and devices concerned. I have not used COPY_N or COPY_H since having TKII.

5.5.1 Single File Copies

This includes the standard COPY command as described above, COPY_N and COPY_H which have also been modified to use default directories, and COPY_O. In my copy of the TKII manual a misprint has lead to COPY_O appearing as COPY_.

The COPY_O command will copy a file without asking what to do if the destination file already exists. This is useful when copying is performed within a SuperBASIC program and one does not wish to give the user the choice of whether to overwrite a file or not.

5.5.2 Wild Card Copies

The command WCOPY allows one to copy a number of files as a single operation. As with the commands for single file copying, WCOPY uses the default directories. The form of the command is:

```
WCOPY #channel, source TO destination
```

As with standard QDOS commands the channel is optional but if supplied it is where the prompts will be sent. If a channel is not specified then prompts will be sent to #0.

The following examples illustrate the use of the command. Assume the data default directory is set to flpl_ and the destination default directory is set to flp2_ and that the disk in drive one contains the following files:

```
tkiaa_doc
tkiib_doc
tkiijobs_doc
letter_RichardAlexander_txt
address_Alexander_txt
```

for all of the examples.

i) WCOPY

This is equivalent to WCOPY #0, flp1_ TO flp2_. That is, copy all files from flp1_ to flp2_. However, as with the WDEL command a prompt is given:

```
flp1_tkiia_doc TO flp2_tkiia_doc..Y/N/A/Q?
```

Responding with A would lead to all files being copied from the first to second disk drive. Individual files may be selected for copying by responding Yes or No as each filename is presented. The operation may be Quitted at any time.

ii) WCOPY #1, tkii_ TO notes_

This is equivalent to WCOPY #1, flp1_tkii_ TO flp2_notes_. Thus a selective copy of only files that are notes on TKII is performed. This would produce the prompt:

```
flp1_tkiia_doc TO flp2_notesa_doc..Y/N/A/Q?
```

The part of the name represented by the wild card is appended to the destination wild card name. For the files that match this specification on the disk in flp1_ the wild card will have in turn the values a_doc, b_doc and jobs_doc. The prompt will appear in #1 ie unless the windows have been changed, at the top of the screen.

iii) WCOPY TO flp1_backup_

This is equivalent to WCOPY #0, flp1_ TO flp1_backup_ and allows copies of all files to be made on the same disk but with a prefix added to the file name ie copy files to a subdirectory.

iv) WCOPY TO ser1

This is equivalent to WCOPY #0, flp1_ TO ser1 and will result in an error: bad name, because ser1_tkiia_doc is not a valid name.

If at any time the resulting destination file exists already a prompt asking if the file should be overwritten is produced.

5.5.3 Background Copying

The command SPL is provided to allow background copying in the same manner as COPY_0. The copying is performed by a spooler which is an independent job. The primary use for the spooler is to print files. SPL uses the data and destination defaults and so if the QL has just been booted then one can print a file as follows:

```
SPL tkiia_doc
```

The command does not accept wild card names. So that a file, flp1_print_cmd (the extension _cmd shows that the file contains a series of commands rather than a numbered SuperBASIC program - the use of sensible and consistent extensions can greatly assist with file management. The extension _bat may be chosen as with MS-DOS) could be created containing the following lines:

```
SPL tkiia_doc
SPL tkiib_doc
SPL tkiijobs_doc
```

The command LRUN print_cmd would then allow the three files to be printed without intervention while the machine can be used for other things. Three separate jobs would be created all name SPL and all running at priority 8. At the default priority the background printing will have little effect on ones main job whether it be editing or playing a game. However, if the destination is a file rather than the serial port this will not be the case. When spooling to a file keyboard response will fluctuate considerably no matter at what priority the spooler is running. Spooling to a file will obviously be much quicker than spooling to a printer but offers no real benefits over

copying to a file.

The output for the spooler is selected using the command `SPL_USE`. This is in fact, the same as `DEST_USE` except that an underscore is not appended to the name - an underscore at the end would indicate a wild card name and `SPL` does not accept wild cards.

`SPL_USE flpl_dump`

Would set the destination default to `flpl_dump` and all subsequent uses of `SPL` would write to that file automatically overwriting the previous version.

A variant of `SPL`, `SPLF` will spool a file and place a form feed at the end. This will ensure that individual files are printed on separate sheets of paper.

Both of the commands `SPL` and `SPLF` may be supplied with channel numbers rather than filenames as explained in `TKII` manual.

At this point it is worth mentioning one of the many wonderful features of `TKII` that I do not think appears explicitly in the manual. Although not directly connected with spooler it is to do with printing.

If, on a `QL` without `TKII` fitted, there are two (or more) jobs running both of which are trying to access the printer the result will be a printout which is a mess - the output of the two jobs interleaved. With `TKII` fitted one job's output will be sent to the printer while the other's is buffered in memory. Once the printer is free the buffered output is copied from memory to the printer!

5.5.4 Renaming Files

As explained in the `TKII` manual the renaming commands follow the same form as the equivalent copying commands but merely alter the filename ie `RENAME` has similar syntax to `COPY` and `WREN` has similar syntax to `WCOPY`.

MODE CHANGE UTILITY

UNFLASH UTILITY

Sometimes it is inevitable that you must load a picture created in `MODE 4` into `MODE 8`. Flashing pixels and lines are generally all that result. One could write a Superbasic routine which gets rid of all of the flash pixels, but such a routine would take ages.

`UNFLASH` is a Superbasic Procedure which can be used to get rid of this problem. The given machine code will load the new procedure.

The syntax is: `UNFLASH` (no parameters)

To enter into a hexadecimal loader, such as printed each month in `QL World` for the Toolkit:

Program length: 54

Dataspace: 0

	Checksum
Code: 43 FA 00 0C 34 78 01 10 4E 92 70 00 4E 75 00 01	:1A
00 10 07 55 4E 46 4C 41 53 48 00 00 00 00 00 00	:28
43 F9 00 02 00 00 32 3C 40 00 02 59 AA FF 51 C9	:0A
FF FA 70 00 4E 75 00 00 00 00 00 00 00 00 00 00	:2C

The utility is also in the public domain library as `UNFLASH_cde`.

(Note this item has been translated by Rich Mellor from an article in `QUASAR 21`

LOOSE ITEMS

ST-QL UPDATE

ROUND THE MAGAZINES

Those of us who read the general computer press will be saddened to see that New Computer Express has dropped its QL column and it looks as if Computer Express, which has carried a regular 3/4 page section on Sinclair computers, including the QL, has gone the same way. Whilst these columns were of variable quality (especially NCE) they provided a point of contact for those QL owners who did not read QL World.

Talking of QL World, I'm given to understand that they have had excellent results from their promotion of the magazine at the newstands over the Xmas period, which should ensure the magazine's continuation for the foreseeable future.

QUANTA has seen some upheaval due to a change in editorial policy (a change not necessarily instigated by the previous editor, Sarah Johnson.) One of the changes is that Dennis Briggs will be monitoring the adverts and technical articles to ensure that nothing gets printed that is misleading, illegal or simply wrong. Dennis has also offered to do the same for us, and we may be taking up his offer relating to one or two items. Another change is that Bill Fuggle has substantially improved the appearance of the magazine. No details of the production process involved.

D-I-Y TOOLKIT

Simon Goodwin has been working further on this series in QL World, and has now completed 14 volumes (or modules), which are all now available from C.G.H. Services. If anyone has any queries about D-I-Y Toolkit they are welcome to write to us, and we'll be happy to pass them onto Simon (or you can write to him at QL World.) I'd be very happy to print any reviews of the Toolkit - whole or segments - in QLTR.

C.G.H. SERVICES POLICY

Over the past couple of years we've built up a good relationship with other QL publishers, retailing their products. We've reluctantly decided that, in order to concentrate our resources, we're no longer selling other people's software. We've not had any problems with other people, it's more a problem with administration and currency exchanging which absorbs a lot of time and energy.

A couple of people have pointed out that Keith Reader's article on converting screen to the QL could have been improved slightly.

Firstly, one doesn't need the st_ql_bas prog in memory at all, only the st_ql_exe and turbo_st_ql. Keep these in a ram disk. e.g.

```
10 FORMAT RAM1_100
20 COPY FLP1_TURBO_ST_QL TO RAM1_A
25 REPEAT LOOP
30 EXEC_W RAM1_A
40 INPUT #0, 'CONVERT ANOTHER PICTURE?
(Y/N)'; A$
50 IF A$=='N': EXIT LOOP
60 END REPEAT LOOP
```

You will of course need a Ramdisk driver either on your disk interface or installed from software to use this facility.

It was also pointed out that Keith chose to convert from medium resolution screens on the ST, when high resolution screens are available and should be better quality. Personally this was the route I took, but then I have a mono monitor for my ST. Not everyone has this and if you are working with a T.V. you will have problems accessing mono screens. Also some ST screens only start out as Low-Res screens in the first place, e.g. Neochrome. If you're converting pictures as opposed to Clip Art, be prepared for a lot of work re-colouring the screens!

We have also taken Keith's screens out of the P.D./Shareware Library and included them in the C.G.H. Services publications list to ensure that Keith gets the money he should for doing all the work on converting the screens. Keith has also added another disk - General 1, and I've done half a disk of Xmas clip art as well.

Alan Pemberton has sent the latest revision of his ST-QL screen transfer program to us suggesting that the program be put into the P.D. Library. The program now incorporates extensive screen manipulation routines. Rich Mellor and I are trying to persuade Alan that this is now of commercial quality and should be distributed as such. Rich has also been working further on the program's screen compression routines, using machine code, making substantial gains in speed.

Richard

FLASHBACK REVIEW

Available from Sector Software or from Dilwyn Jones Computing
Price £45.00

Arthur Miller objected to the term "flashback" to describe what happens in his play "Death of a Salesman"; he preferred to say it contained a "mobile concurrency of past and present". The same applies in some ways to Sector Software's "FlashBack" database, as compared with the Psion "Archive". "FlashBack" is fast - very fast - partly because it seems to be an excellent piece of programming, and partly because it holds all its records in memory. There is no laborious accessing of microdrives or disks to slow up record retrieval. The earliest and latest records cohabit. Past and present are mingled.

Of course, the system has its disadvantages. You would not want to hold a really enormous database in memory, so "FlashBack" is better for relatively short databases - say about 500 records, maybe? Though, of course, a lot depends on how complex your records are. Then again, there is nothing to stop you splitting your records into two batches: I have known "Archive" take longer to find a record than it takes "FlashBack" to load and review a complete file.

Once your file is in memory, you can scroll through it in brief form very rapidly indeed, locate a record in split seconds, add or amend records without any hassle at all. To revise a record, you just type in an extra bit, or delete what you don't want - there are no fixed-size fields to worry about. You can change field names, add new fields if you wish, or split existing fields, whenever you feel like it. You don't have to worry about opening or closing files as you do in "Archive", so you don't need to bother with any recovery system. Nothing much is likely to go wrong; if it does, you can easily correct it. Sounds like a dream? It is, compared with "Archive".

Of course, "FlashBack" is simple. You can't make up fancy screens as you can in "Archive", and the picture on screen is very functional. You can, within reason, have the "FlashBack" screen any size you like; when you save your database, the screen size will be remembered when you reload. (Using a small screen saves memory, if storage space is a problem.)

The package contains a routine for turning "Archive" files into "FlashBack" ones; the immediate effect to cut the size of the data file drastically. If you do this with the "Quanta" library file, you will find yourself dealing with a much less pretty screen, without the elegant layout of the original package. You will also find yourself referring to the library a lot more, because it suddenly becomes so quick and easy: you can probably browse rapidly through the whole Quanta library in ten minutes or so, instead of gnashing your teeth, or wondering whether to make a cup of tea while "Archive" fails to find what you want.

There is, admittedly, no sophisticated programming language in "FlashBack". I wonder how many "Archive" users make use of this facility, though. For those who require a simple card index system, "FlashBack" is much superior. You have only about twenty commands, accessible by F3-Letter or by CTRL/letter. For example, CTRL/v throws up a pageful of abbreviated records to scroll through; F3 followed by "v" does the same. CTRL/s (F3-s) searches for a string. CTRL/g (F3-g) groups records by topic for viewing or printing. There are three different methods of organising the records alphabetically (CTRL/i or F3-i). All this, you see, is easy for the average idiot to comprehend and use with very little reference to the manual. For the really forgetful, you can screen all the commands by dabbing F3 twice, and access whatever command you want with the scroll-bar and [ENTER].

"FlashBack" has its subtleties. There is, for instance, a second screen available, which uses sub-record markers in a slightly different way from the initial screen. I have had no need for recourse to this facility, and haven't yet bothered to read the manual thoroughly on the subject. The database can obviously do simple things like names and addresses. Less obvious is the sort of use to which Sector Software put it:

the manual is itself a database, with a clever navigational system, and it shows that "FlashBack" could be used, at a pinch, as a perfectly good word processor.

While we are on the subject of manuals, the one I got with "FlashBack" (ordinary - I don't know if Sector Software still do a printed manual with the Special Edition, since all the instructions are contained in a database file) is mercifully short. Certain other praiseworthy software producers, who shall remain nameless because I like them so much, could take a lesson from this! - Length is not necessarily equivalent to clarity.

The keynote of the whole system - manual, "FlashBack" and the Report Generator, is simplicity and ease of use. There is plenty of flexibility, too, but the emphasis is on getting the user up and running with the minimum of fuss and the minimum expenditure of time.

Grouses

So what are the drawbacks? Yes, there are a few. I have already mentioned one or two that do not matter to me. Those that do:

1) The command to create a new record is F3-c or CTRL/c. CTRL/c?... But isn't that what you use to switch jobs when multitasking? Unfortunately, yes. Bit of a blunder here, Sector, I feel. CTRL/+, perhaps? In effect, then, if you are multitasking, you can use only the F3 system or the scroll-bar system to create a new record. The Special Edition does multitask, if not perfectly happily. I use it with QPAC2, and have used it with QRAM - but you have to make sure to access "FlashBack" from Basic.

2) There is a long and involved "boot" program in SuperBasic - much longer than it needs to be. I have dispensed with it, for almost all purposes.

3) There are two commands for tracing strings. CTRL/g "Groups" records by reference to a particular field and string, or more generally. CTRL/s "Searches" for the next occurrence of a specified string, starting from where you are in the file, and searching either a specified field, or all fields. A general search of all fields, however, will find a string which occurs elsewhere than at the beginning of a field. Otherwise, if one field consists of "Kafka, Franz", you can successfully Group or Search for "Kafka": but you cannot trace "Franz".

4) For reasons I do not understand, the Report Generator makes up its own device name, and depending on what junk you have in your machine, the device name - unalterable at one significant point - may give either a tidy "flpl_" or whatever, or a row of inaccessible blodges. This may have been amended in later versions of the program. In my case, a phone call to Sector Software encouraged a nice man there to tell me about an amendment to the boot program which solved the problem.

And that's it. Four is the sum total of all the grouses, gripes and groans that I can think of. Not bad, I feel.

It may be helpful to some users if I mention a solved problem, too. Special Edition originally produced, when I used it with the Minerva ROM, a "hyperactive" cursor. I wrote at once to both Sector Software and QView. Two days later I got a phone call from Stuart McKnight who explained the problem, and said he was sending back my copy of the program with an amendment. The program arrived the following day, but failed to load; fortunately, enough of it did to enable me to see what alterations he had made, and perform them on the whole program. The cursor became magically normal. (Months later, I received an undated and useless response from Sector, suggesting that there must be some bug in my setup, and that nothing could possibly be wrong with "FlashBack"! - I wrote again, to say that Stuart McKnight could help them. BUT see above how helpful Sector can be on the telephone!)

Report Generator

The Report Generator is part of the package. It takes time and a modicum of

concentration to sort out how to use it, but if I can do it, so can anyone else! With this program you can print out your records in whatever way you decide, leaving out some fields, if you wish, printing fields continuously or on separate lines. So it is easy enough to print out a whole series of address labels, for instance. Or you can use the system for "Mailmerge".

USES FOR "FLASHBACK"

What do I use "FlashBack" for?

- Keeping a record of articles and programs for the QL - a form of incest common among QL users.
- Similarly for Z88.
- Maintaining the Quanta Library.
- Keeping a list of names and addresses of customers, and printing out such as are required with the Report Generator.
- Keeping a list of names and addresses of institutions with which I have business dealings, and printing labels as necessary with the Report Generator.

There are various other odds and ends, too - but on the strength of these uses, I think "FlashBack" has fully justified the outlay of £40 to Sector Software (0772-454328/452414, credit cards accepted).

What do I use "Archive" for?

Nothing.

(I know nothing of Digital Precision's "Qflick", and would be interested in a comparison between it and "FlashBack" by someone with both.)

Mike Edwards.

In response to Mike's review, above, I have received the following reply from David Batty of Sector Software:

"Using Ctrl C as one of the keys in Flashback does not stop anyone from using Flashback in a multitasking machine. We use it in this form and can still multitask other programs at the same time.

The device name is obtained from the boot program and will always be correct as in FLPl_ or MDVl_ etc. This is unless you have altered the boot program and taken it out of there. There was an error in the first 30 to 50 copies of Flashback boot program that gave the wrong device name, this was soon corrected and patches given to customers with the faulty boot program."

Whilst we're dealing with Sector Software, David has moved back to his old address at 39 Wray Crescent, Ulmes Walton, Leyland, Lancs, PR5 3NH. This move is designed to give David the time to do more programming and avoid constant interruptions. In this I wish him well, although if my home is anything to go by all that happens is that the time at which you can be interrupted extends beyond shop hours to up to 10.00 pm or even later.

We should also thank David for organising the Northern Computer Shows in Leyland as they have provided us with opportunities to meet other QL users in the area and make more sales!

Next issue Mike will be reviewing PROGS excellent DataDesign database. Later issues may well feature a head-to-head between QL databases.

QMENUS

The following piece concerns the menu extensions by Jochen Merz.

Menus is given away with QDII and is an extension thing. It provides utility routines available to any programmer (not just SuperBasic people).

PROCS

TH_MENU

Enforce the Thing definition of the extension thing 'menu'.

REPORT_ERROR error_code,x,y

Equivalent to a tk2 REPORT command in a pull-down window.

VIEW_FILE filename

Easy way to show a file to the user.

BUTTON_WAIT

Set up a button (not quite in the button frame) and wait for it to be pressed.

FUNCTIONS

FILE_SELECT\$('title', 'filename', 'directory', 'extension')

LOAD FILE_SELECT\$('Load a graphical doodle', 'flpl_Vis_Flowers_bas', 'flpl_Vis_', '_bas')

Get a filename from the user. Very nice. It can handle hard sub-directories like Qpac-2 but not soft directories.

DIR_SELECT\$('title', x, y)

Select a directory from a data thing that has a list of directories in it - this default thing is another Jochen Merz special.

EXT_SELECT\$(x, y)

Select a file extension.

READ_STRING\$('title', 'default text', 'prompt', x, y)

Get the user to edit a string.

Sat 15 Sep 1990 2:09:00

Well, I must go now, its late

Ian.R.Bruntlett

(Ian provided us with seven screen dumps which, whilst quite useful in some respects were 95% white space, and space being at a premium there were better things to print in this issue. Maybe we'll do a disk like the QITALY ones with plenty of screen dumps and similar (perhaps with some PD progs thrown for good measure!)(Or maybe Ian will get QL Forum going again?) Let us know if you think this would be a good idea. We could add a quid to each issue's cover price to make it viable.

QMenu, published by Jochen Merz, Im Stillen Winkel 12, 4100 - Duisberg 11, Germany, costs a mere £12.10 (all incl.) Jochen's latest advert in QL World also mentions a new upgrade to QD as well. We'd really like to have some proper reviews of Jochen Merz's software and hardware products.

Richard.

THE PAINTER - REVIEW

Published by Progs
and available from Haachtstraat 92, 3020 Veltem, Belgium

The advent of home computers has produced a complete new format for artists to portray their natural talents. The traditional tools of oils and canvas, watercolours, pen, coloured pencils and paper have been joined by the coloured pixel on a TV/monitor screen and a dot matrix printer.

The block graphic symbols on the Sinclair ZX81 which could be positioned on screen using the keyboard to produce a picture have rapidly been superseded by sophisticated art programs which enable the computer user to produce impressive coloured masterpieces on the screen. The use of a colour printer enable the picture to be transferred to paper it is preserved for posterity.

Several packages have been available for the QL since it's invention the most famous of which is probably the "Eye-Q" program from Digital Precision. I must admit that I use this program although I do not think it lives up to the image claimed by the D.P. advertisements. I have seen more impressive art packages on the Spectrum.

An art package which has been available for the QL since 1988 has now been updated and improved. The Painter published by PROGS (PRO fessional & Graphical Software) and produced by the Belgium authors Joachim and Nathan Van Der Auwera is now readily available in this country.

The program which comes on a 3.5" disk requires at least 128K memory expansion. I used it on a JS ROM with a 896K Trump card. Although the program loaded and appeared to run satisfactorily on my Minerva ROM I found that the machine crashed when I tried to generate additional screens.

The manual supplied is very comprehensive although because it has been produced by the authors contains some very quaint English and is therefore, in places, difficult to understand.

The program is designed to be used in conjunction with Qjump's Pointer Environment which includes a mouse. I do not own this system so I cannot comment but I used it very successfully with an ordinary joystick. It will not work with the ICE Mouse.

I successfully multitasked it using Qram. In fact it is in the computer as I write this article with Quill.

Upon loading you are confronted with the main menu page which is very colourful. The program boots in mode 4 and hence there are only 4 basic colours to use. The mode can be easily changed to give 8 colours which are shown on the colour palate in the upper half of the menu screen.

The menu is split up into five sub-headings Shapes, Corrector, Screen Adds, Colour Manipulation, and Others. Further options are Files, Dump, Quit and Help whilst colour selection can be chosen from the colour palate mentioned earlier.

Switching between the menu and the drawing screen is via the <ENTER> key and it is suprisingly quick with the screen drawing almost instantaneously.

All the standard methods of drawing are included i.e. square, rectangle, circle, ellipse, arc and with elastic line drawing as an important feature. A useful touch is that you can turn on the "filled" option and as each figure is drawn it will be filled by a colour of your choice. The paint option allows this to be carried out at any other time.

Chosen sections of the screen can be enlarged to enable individual pixels to be switched on or off. A facility enables any shape to be turned to any new angle. Whilst

any selected section of the screen can be enlarged or shrunk according to choice.

I do not have the space to detail all the numerous features of this program so I will just highlight some of the more unique options. I should at this stage admit that there are some features I have not yet mastered to my satisfaction!

One of the most impressive features of the program is it's ability to work with up to 12 screens with the facility to be able to copy one screen to another. An interesting option here is the filter feature which enables two screens to be merged with the current ink colour not copied. I must admit the manual is somewhat vague on this item and I have not fully understood the use of it but it would appear to be similar to the feature used in television whereby a colour usually blue is chosen and provided the same colour is not used in the main picture it is possible to superimpose a backdrop. For example a raft in the studio can be made with a suitable backdrop to appear it is at sea.

There is a very useful fill/brush feature with a choice of different patterns within the program. There is an option to modify the existing patterns or design your own particular ones. There are 24 patterns already in the program ranging from our old friend the brick wall to a set of grinning faces.

As well as the 8 basic colours available in mode 8 there is a separate palette screen whereby the stipple colour feature can be utilised. By a system of pixal configuration the stipple pattern can be changed I estimated that with this facility a total of 256 different colours can be produced and used in the program. These can all be used as the ink or paper colour or under the fill facility.

There is a complete section on text handling which runs via two menu screens. Text can be up to C-size 9,9, Shaded, Outline, Block, Italics (Sloping at various angles backwards and forwards) and Underlined. It can be printed proportionally, sideways, vertically, rounded, bold and inverted. Up to 18 fonts can be worked with and there is also the facility to design your own. The text options seem endless and there are far too many to be able to check all combinations in a simple review of this nature.

The defaults enable the mouse/cursor reaction to be altered for speed, "wake-up" speed, acceleration and pause. The output channel can also be changed so that files can be written to disk. Also 9 pin or 24 pin printer options can be chosen.

Files has the usual Save, Load, Delete, Directory options with the added facility to save the screen in the compressed format. A saving of about 22K on the normal 32K screen.

The dump option enables the screen to be printed with, as usual in this program, several added features; size, grey shaded, part screen, invert, (this means inverse), whole screen sideways and half screen whole page. The dump facility worked on my Star LC10 printer without any alterations to the printer driver.

I consider that this is the most comprehensive art program published for the QL and it is difficult to suggest any further features that can be added. I was particularly impressed by the multiple screen facility and the text handling. The ability to use all it's facilities correctly will require considerable time and effort but this in no way inhibits the novice in producing satisfactory results from the moment the program is loaded. It is a program that I can fully recommend and at a selling price of only £30 it is, in my view, an absolute bargain.

Keith Reader

STOP PRESS

Have just received a letter from Joachim and Nathan saying that they have now upgraded The Painter to Version 4, with a recommended retail price of £45. I have yet to ascertain whether Version 3 is still available at the price stated above.

CLIP ART REVIEW

A Review of the Clip Art Disks published by PROGS
available from Haachtstraat 92, 3020 Veltem Belgium

With the increased availability of Desk Top Publishing, with at least four programs available for the QL, there has been an increasing need for "clip art" so that publications can be illustrated.

Unless you have artistic talent and the use of a good art program such as The Painter it is difficult to produce pictures for your Newsheet, Poster or whatever. Clip-art is a selection of ready made pictures or small designs already copied onto a disk as a series of screens. These can then be loaded into a DTP program and the relevant picture chopped out or into an art package and alterations made to the picture if these are desired.

'Progs' the trade name for the software produced by the Belgium authors Nathan and Joachim Van der Auwera have now issued three disks of clip-art suitable for use with their art package "The Painter" or, after suitable modification, for using in QL Desk Top Publishing programs.

The clip art screens come on 3.5" disks and because all the screens on the disks have been compressed 152 screens are found on the three disks. The compressed format is suitable for loading into The Painter but to use them in any of the desk top publishers for the QL would require the screens to be re-saved as normal Sbytes 32K screens. As an added bonus but this is only relevant if you have purchased The Painter there are six extra fonts for use in this program on the third disk.

To enable the contents of each disk to be seen on the screen without loading into an art program each disk has a small "Boot" program on it which when implemented loads the screens and displays them in turn. This is a very useful utility.

The three disks come complete with a full manual which gives instructions how to load the compressed screens in SuperBASIC and how to save them in a normal screen format. There is a full list of the screen names to be found on each disk with a list of each picture on individual screens. The latter part of the manual consists of a list in alphabetical order of each picture to be found on the disks. The screen name is given and the number of the disk on which it is to be found. This index is extremely useful and adds a nice finishing touch to the program presentation.

I must just mention the title of one picture in the index which particularly intrigued me. "Balance-cord" was the name used and I quickly loaded up the screen to find out what this was? The answer a "tightrope walker". Having said that I wish that my French was even a quarter as good as the authors.

Now down to the nitty gritty; what exactly do you get for your money of £30 for the three disks? Well the quality of the clip-art varies, in my opinion, from excellent to down right disappointing. Clip-art screens can be drawn originally using an art package and whilst this, with a good artist, produces an excellent sharp image the quickest way to produce screens is to use a scanner saving pictures from a book or magazine. This latter method can produce very variable results.

The type of pictures offered ranged through scientific signs, aeroplanes, animals, sports players, cars and trucks and furniture to name but a few. It is a pity that the quality of the three screens depicting the QL computer are so poor.

The series of Art screens on Disk 1 are of a very high standard producing clear black and white pictures whilst the series of Car screens on the same disk appear to have been scanned and are particularly poor with the image consisting of black, white, red and green on my colour monitor. The latter screens printed very poorly. All the screens on disk 1 contained several small pictures but on disks 2 and 3 some screens consisted

of just one large picture. Quality also varied enormously on these disks. I thought that the standards of the clip-art deteriorated as one progressed through the disks.

The clip-art disks were well presented with a useful manual but I consider that the contents of them were somewhat disappointing. I think that the poor quality screens should have been omitted and consequently the price reduced. In the case of this product I feel that quality has suffered in the pursuit of quantity.

Keith Reader

COCK-UP CORNER

Traditionally we have featured this column to point out problems in software, manuals etc. However this time we lead off with errors spotted in the last issue of QL Technical Review.

Firstly thanks to Dennis Briggs for spotting that I inadvertently called the QUEST disk operating system software, the MEDIC one in the Disks and Disk drives section. So, to clarify, Michael L. Jackson, who was not responsible for the error, has revised the QUEST software, not the MEDIC one. As Dennis correctly pointed out, the MEDIC boards worked O.K. when fitted with the v1.18a ROMs, which have the earlier bugs removed. Please note that the description of the revision is correctly named in the CGH Services PD Catalog (which I also wrote!)

Secondly Dennis mentioned that we gave the impression that Keyboard Products Ltd had taken over the Schoen product line. What I actually tried to say (but obviously failed) was that Keyboard Products Ltd are now marketing their own keyboards. These are the keyboards previously sold by Schoen. Dennis also points out that they should not be described as PS/2 or XT keyboards but as PS/2 or XT STYLE keyboards. My apologies once again.

Dennis also wrote us a long letter several months ago after QLTR 1 but due to a variety of factors it only reached here after QLTR 4 was put to bed. Not wishing to wake the last issue I had intended printing it in this issue but there is so much material waiting to be printed that it will have to wait, perhaps until next time.

LOOSE ITEMS

Amongst Dennis Briggs' copious notes that he has recently sent us he mentioned a few items that may be of interest to readers.

BT PLUGS

Dennis has noticed that QL owners frequently carve BT plugs to make them half fit the serial ports. BT plugs are PTC plugs, whilst the QL needs PCC plugs. The reason for the difference is to stop computers with PCC plugs being plugged into BT sockets and having BT screaming down your neck for wrecking the phone line. Dennis (who trades so successfully as Adman Services that he does not need to advertise his wares) has had the PCC QL serial plugs in stock for three years. So if you need plugs get the right ones from Dennis.

SPEEDSCREEN

Simon N. Goodwin has kindly allowed C.G.H. Services to sell off the remaining stock of Speedscreen ROMs's and flp/mdv's. Price will be £20.00 for the ROM version (included in this price is a mdv/flp - please state preference). For the flp/mdv version the price is £10.00. All prices subject to 10% surcharge for postal sales.

UNOFFICIAL THOR USERS' MAGAZINE

Malcolm Smith has sent us a copy of the first issue of MJOLER, the Unofficial THOR User's Magazine. It's an 8-page A5 magazine and should build into a useful resource for THOR users. Malcolm concentrates on a few of the problems, THOR users face, including that of compatibility and inability to use programs using mdv_ keys. One query that Malcolm raises is that the current status of THOR International is unknown. Contact Malcolm Smith at Statsrad Ihlsensvei 66b, N-2010 STROMMEN, Norway.

BROTHER HR5 UPDATE

Chris Adams

George Jones (QLTR 4) doesn't need to write a fancy screen dump program for his Brother HR5 - he's already got one. Easel's gprint_prt file works fine with the HR5 from Superbasic:

```
a=respr(510) : lbytes flpl_gprint_prt, a : call a
```

This assumes you've got the gprint_prt file from Easel in flpl_ of course. It seems to skip 16 pixels from both left and right of the screen - perhaps because Easel graphs never use the full screen width. Easel's default 'P' option to print a graph also works with the HR5, which is sort-of Epson-compatible. The dumps are inverse - white prints as black and vice versa, blue is a light grey-scale and yellow a dark one, etc. (Incidentally, Easel's grey-scaling of screen colours is still the best I've seen. Has any other program bettered it in the last six years?)

All the graphics progs I've used can be configured for the HR5 - Eye-Q, Professional Publisher, the Quanta library version of Page Designer, Sidewinder Plus, and others.

The printer has two graphics modes - the standard bit-image ESC K, which prints a maximum 480 dots per line, and double-density ESC L, which prints 960 dots.

If you've selected the UK character set with the dip switches to get a £ sign using Quill printer translates, you won't be able to print a hash (#). A way out is to use another translate that prints # using bit-image mode. Run the Psion install_bas program, select your printer driver from the opening screen and key F2 to edit it. Pick an unused Translate line if you've got one and enter:

```
TRANSLATE10      :35,27,75,5,0,72,254,72,254,72
```

When you press ENTER the figures will disappear and return looking like this: #,ESC,K,ENQ,NUL,H,254,H,254,H. When you've finished press F5 to install the driver. Now, when Quill sees a # sign it will send codes to the printer to turn on bit-image, print the hash sign and go back to text-printing mode to continue with the document. To see how it works read page 35 of the HR5 manual (ESC K) and look at page 47 (dot patterns).

If you don't do this a line in a Quill doc that says: "OPEN_IN #3, flpl_test" will appear as "OPEN_IN £3, flpl_test". NB: Don't use this printer driver to produce Superbasic listings via a _lis file, in fact don't use a printer driver at all.

You can use this 'Quill' printer driver to print hashes in Archive and Abacus too, of course.

The HR5 has a number of in-built graphics characters. Those from character-code 224 onward can only be printed by sending the code (eg try: OPEN #3, ser1 : FOR n=224 TO 254,10,13 : PRINT #3, CHR\$(n);" "; : END FOR n : CLOSE #3). You'll need printer translates to use these in Quill.

However, the character codes from 128 to 159 can be sent to the printer using QL keyboard equivalents, so you could put, say, the funny character on CTRL-8 in a document and it would print as a top-left box corner. CTRL-SHIFT-7 draws a solid square, etc. The table on page 14 of the printer manual lists the characters and their codes. There's a warning that it won't work with RS-232C 7-bit systems, but my HR5 is RS-232C like George Jones's, and it works fine. The QL sends 8-bit data bytes to ser1 unless you arrange things otherwise.

The one-time carbon fibre ribbons can't be re-inked. You can buy them from Brother itself for £3.40. Ask for 6040 HR5 ribbons, product code 00302. Address: Jones + Brother, Shepley Street, Audenshaw, Manchester M34 5JD. Tel 061-330 6531.

MANDELBROT UPDATE

PD. NEWS

A couple of items of interest on the fractal front.

The Amiga QL emulator by Rainer Kowalick has, amongst its QL progs, one called Box_bas. Not a very interesting name, but it hides (or rather opaquely discloses) a good Mandelbrot generator program. It does however suffer from the usual collapse into incalculability after only 4 or 5 zooms. Bob Macey has taken this program and added a couple of lines which transform it into a program capable of producing quite beautiful pictures, using a jewelled effect. Brilliant.

Secondly PROGS, producers of The Painter and DataDesign, have announced a new Mandelbrot and Julia program entitled Qractal. The blurb claims fast and accurate calculation, use of Binary Decomposition (?), Inner Set Filling, special colouring methods and much more. I'll let you know more when we've been able to review it.

If your are interested in Fractals I can thoroughly recommend John de Rivaz's publication "Fractal Report". A six issue subscription (one year) costs £10 (UK), £12 (Europe), £13 (elsewhere). Cheques payable to Reeves Telecommunications Laboratories Ltd please. The latest issue, #12, has 20 pages and has articles on speeding up fractal generation, spirals, hyperbolic patterns and recursion, wolf's dust amongst other things. The editorial also keeps one up to date on a variety of fractal related products, although few, if any of these are for the QL. If you have managed to convert any of the programs to SuperBasic do let us know. I tried to convert a SAM Coupe basic prog but failed totally. John has given permission for us to put progs into the PD library.

If you are into the weirder extremes of fractals, virtual reality, designer lifestyles, neuropolitics, etc I can thoroughly recommend a new American journal called "Mondo 2000". Any mag that can bring together Timothy Leary, Robert Anton Wilson, CyberPunk, The Church of the SubGenius, The Grateful Dead, Virtual Realities, Acid House and Camper van Beethoven can't be all bad. Cost is about £4.50 depending on your supplier. I may be able to get copies for people.

Richard Alexander

It's difficult to remember exactly what has been added to the C.G.H. Services P.D. and Shareware Library since last time. As usual check the enclosed price list and order form for prices and availability.

We've been kindly sent by the Scottish QL Users Club a disk of their P.D. software which includes some Abacus spreadsheets and numerous utilities.

Emmanuel Verbeeck has updated his P.D. disk, with new programs such a disk cataloguer, super booter, tools, super kit merger etc.

Oliver Fink's QLaboratories disk has come our way. This concentrates on utility programs that use the extended environment QPTR programs. Included are a Font Editor and a prog to help programs make use of the extended environment. Ian Bruntlett has kindly added a few comments on the disk.

Also recently arrived are two Shareware titles: QED by Jan Bredenbeck and Harvey Taylor's QLINK program. The first of these two is a text Editor, whilst the other is a comms prog from Vancouver. We await comments on these from users.

The educational P.D. collection slowly builds, with Bill Cable's excellent maths learning progs from the States and Ian Thompson has updated the Periodic Table of elements to incorporate a bug-fix.

We've also had two disks of Australian software from their QL User group. We're still checking through these. Most of the progs seem to be games but there look to be some interesting utilities. We'll weed out the QUANTA and QL World stuff first.

We've also just got some Stephen Michels disk utility progs which look very neat. I'll be checking these out as well.

We now have several of the QITALY disks. These feature program reviews, as well as articles. Many of the items on the disks are screen dumps from programs, so even if you don't speak Italian, you may find the disks of interest.

Finally Bruce Nicholls' Star LC-10 Printer Demo prog which complements his article also in the library. More printer progs and other progs are always welcome!

Richard Alexander

QL TECHNICAL REVIEW #5 UPDATE SHEET

Since committing the bulk of number 5 to paper, ready for printing out, we've been beset by a few problems. Nothing unusual there, of course! A pleasant problem was the huge influx of orders in the pre-Xmas period, which, combined with several successful stalls at computer shows and workshops, meant that the time remaining for the production of QLTR5 was dramatically curtailed. Unhappily we were also on the receiving end of a series of hardware failures. It is suspected that the weather was to blame for some of them - our already suspect mains supply was even more erratic than usual - and our office was so cold that one of my QL's simply refused to boot-up! At various times we were without an expanded QL, a disk interface and two 3.5" disk drives and have only just got back to normal a month after the hardware went haywire. The replacement and repair bill (in particular the new disk drives) was a most unwelcome surprise. Inevitably we have fallen behind with several projects due to these setbacks but are slowly catching up with them.

ERROR REPORT

One of the casualties has been the Error Report for QL Technical Review to keep these up-to-date and correct previous mistakes. Should be ready for #6, all being well.

DIY TOOLKIT

Please ensure that if you are ordering modules from DIY Toolkit, that you make any cheques etc payable to DIY Toolkit, and not to Simon Goodwin or C.G.H. Services.

POLYTEXT

We are now supplying this. The first batch proper went out just after Xmas and apart from refusing point blank to work on a THOR, we have yet to have any feedback, positive or negative. As the program is designed to enhance your Quill output via your printer we're anticipating that some people may have problems, but they haven't materialised.

QL TECHNICAL REVIEW 6

As this issue will be with you by the end of January, #6 should be out by Easter 1991 as there is quite a backlog of material to print. Items for #7 are most welcome. We shan't be investing in further QL hardware or software here (apart from keeping our existing set-up working!) so please submit material for printing as Quill_doc files, and if you can put your text through a spell-checker before sending it, so much the better. We'd appreciate it if people could check their comments with publishers/suppliers before sending them in.

QL TECHNICAL REVIEW 5

In the time taken to put the issue together, and notwithstanding the numerous changes we made after the initial layout, there are still some things that are out-of-date. One of these is that Digital Precision have dropped the price of Turbocharge + Toolkit to £79.95. The name of DP's new (I hesitate to use the term) "word processor" is Perfection and costs £119.95 when bundled with a spell-checker, £79.95 on its tod. In case you miss it, C.G.H. Services are no longer selling other people's software. This is to save us time and hassle, more than anything else. So please, no more orders for software from other publishers. The last casualty of this policy is "War In The East MKII" which we were hoping to sell. Looking at the photocopies - it seems the copier has missed off some of the page numbers - apologies all round. Even the first run of these were badly faded!

Right, that's your lot
Richard Alexander,
C.G.H. Services.